Entrada e saída de dados

APRESENTAÇÃO

A tarefa de um programa de computador é receber dados por meio de instruções de entrada, processar essas informações de acordo com a lógica criada e traduzida para uma determinada linguagem de programação e entregar esses dados processados utilizando as instruções de saída de dados.

Nesta Unidade de Aprendizagem, você irá estudar uma parte importante do desenvolvimento de sistemas: as instruções necessárias para realizar a entrada e a saída de dados em um programa de computador.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Definir função.
- Aplicar as funções printf() e scanf().
- Desenvolver códigos de formatação.

INFOGRÁFICO

Uma função é um subprograma que só é executado quando chamado, evitando repetições, ou seja, podemos pegar esse trecho de código que irá se repetir e colocar dentro de um bloco - uma função.

Outra característica é que as funções ajudam a dividir um programa, deixando o código mais organizado. Cada função executa uma operação predetermina. Existem funções predefinidas como: printf () e scanf(), e as funções definidas pelo usuário.

Uma função pode ser pensada como uma caixa-preta, quando se utiliza uma função não é necessário "estar a par" de todo o código existente dentro dela, apenas saber o que precisa entregar para a função (se precisa) e o que ela vai entregar (se entregar).

Acompanhe no Infográfico como a caixa-preta funciona, arrastando a seta.

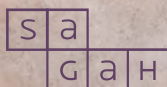
Conteúdo interativo disponível na plataforma de ensino!

CONTEÚDO DO LIVRO

Os computadores precisam de dados, que são processados pelos programas, para executar as suas tarefas. É preciso que nos programas existam instruções para que esses dados sejam inseridos e, além disso, instruções para que a resposta desse processamento seja mostrada. Essas instruções são as instruções de entrada e de saída.

Leia mais no capítulo Entrada e saída de dados, da obra *Algoritmos de programação*, que serve como base teórica para esta Unidade de Aprendizagem.

Boa leitura.

ALGORITMOS DE PROGRAMAÇÃO

Marcela Santos

Entrada e saída de dados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever o conceito de função.
- Aplicar as funções printf() e scanf().
- Usar códigos de formatação.

Introdução

Os programas de computador são sistemas que processam dados. A tarefa de um programa de computador é receber dados por meio de instruções de entrada, processar essas informações de acordo com a lógica criada e traduzida para uma determinada linguagem de programação e entregar esses dados processados, utilizando as instruções de saída de dados.

Neste capítulo, você estudará uma parte importante no desenvolvimento de sistemas: as instruções necessárias para realizar a entrada e saída de dados em um programa de computador.

O conceito de função

Para explicar o conceito de função, vamos começar com o seguinte código:

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("Bem-vindo ao seu primeiro Jogo em C!\n");
5      return 0;
6  }
```

Figura 1. Código-fonte: o conceito de função.

Na linha 4, temos o uso de uma função existente em C, a `printf`. Esta função escreve na tela o texto existente em parênteses e entre aspas. Existe uma complexidade gigante para que seja escrito na tela do computador um valor e existam instruções específicas para isso.

Fique atento

Uma **função** é um conjunto de instruções que realizam uma determinada tarefa. Para que fique mais fácil o uso, elas têm nome.

Seria preciso um conjunto dessas instruções todas as vezes em que essa operação fosse necessária em um programa. Para facilitar esse trabalho, em programação, existem dois conceitos muito importantes: biblioteca e funções.

Na vida real, também temos algo parecido. Imagine que você tenha que fazer café. Automaticamente nós já realizamos todas as tarefas que estão envolvidas: colocar água para ferver, colocar o pó do café, esperar a água ferver e despejar a água quente sobre o pó de café (essa é a maneira que eu faço café!)

Então todas as vezes que eu vou fazer café, mando essas informações automaticamente para o meu cérebro. Em programação, é a mesma coisa: uma função é um conjunto de instruções organizadas em bloco, que possuem um nome e, quando for preciso, é chamada por meio deste.

As funções podem ser criadas pelos programadores ou podem estar incluídas nas bibliotecas de uma determinada biblioteca — e é aqui que entra o nosso segundo conceito: uma biblioteca de programação é um conjunto de funções que já foram pré-definidas por outros programadores; você precisa simplesmente adicionar a biblioteca ao seu programa e utilizar as funções.

Esse é o caso da função `printf`, presente na linha 4 do código apresentado na Figura 1. Agora, vamos aprender mais sobre essa e outra função, que são as funções básicas de entrada e saída na linguagem C.

As funções printf() e scanf()

Um computador é uma máquina que faz uso de programas e dados para realizar tarefas. Esses programas recebem esses dados, processam e entregam-nos como resposta a algum tipo de problema. Os programas são compostos de instruções que fazem esse processamento. Essas instruções podem ser funções de entrada e de saída, e é sobre elas que iremos tratar.

Vale a pena ressaltar que estamos tratando da linguagem de programação C, e essas funções sãos funções desta linguagem.

Fique atento

Para usar as funções scanf e printf, é preciso incluir a biblioteca stdio.h no código-fonte, por meio da linha

```
#include <stdio.h>
```

A função `printf` é a função de saída de dados no vídeo. Todas as funções têm uma sintaxe, que é uma regra que mostra como ela deve ser usada e qual o resultado esperado. A sintaxe de `printf` está representada na Figura 2, a seguir:

Figura 2. Sintaxe da função printf.

Essa é a forma mais básica para utilizarmos essa função, quando queremos simplesmente apresentar na tela uma mensagem de texto. Mas pode haver situações em que queremos apresentar o valor de uma variável. Quando isso for necessário, basta que, em vez do texto, coloquemos somente a variável sem as aspas.

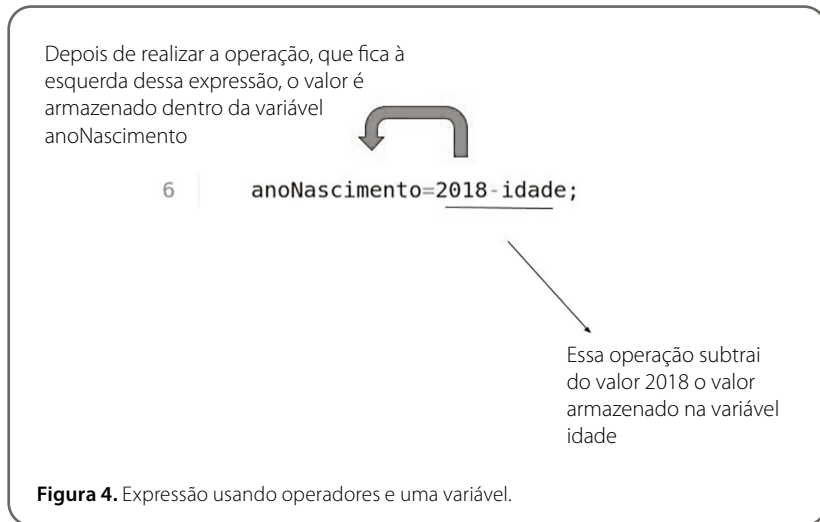
Mas ainda pode existir a situação em que queremos apresentar um texto e uma variável na mesma saída de dados. Para entendermos como isso pode ser feito, vamos analisar o seguinte código, na Figura 3:

Figura 3. O uso de printf com código de formatação.

Esse código mostra o ano que a pessoa nasceu, usando, para isso, uma variável que contém a idade da pessoa. Foram definidas 3 variáveis do tipo inteiro: idade, anoAtual e anoNascimento.

As variáveis idade e anoAtual possuem valores iniciais, ou seja, não temos uma entrada de dados por parte do usuário que executa o programa criado com esse código-fonte. A variável anoNascimento é calculada usando-se dois operadores: o de atribuição e o de subtração — esses operadores serão vistos em capítulos posteriores.

Nesse primeiro momento, vamos entender o que a linha 6 faz, conforme representação na Figura 4:

Figura 4. Expressão usando operadores e uma variável.

Com o valor armazenado na variável anoNascimento, falta realizarmos a saída de dados, ou seja, escrever esse valor na tela. Isso é feito na linha 7, utilizando-se a função `printf`, só que desta vez essa função é usada com um código de formatação e a concatenação de uma variável.

Concatenar uma variável é unir variáveis e/ou textos na mesma saída de dados, por exemplo. Para isso, adicionamos, ao final do texto que deverá aparecer na tela, uma vírgula (,), seguida das variáveis. Cada variável que é adicionada tem um tipo, e é preciso "avisar" ao `printf` que tipo é esse, para tal, o C tem o conceito de códigos de formatação. Mas antes de entendermos o que sãos os códigos de formatação, vamos à nossa função de entrada de dados.

Bom, agora vamos para a próxima função, a `scanf`. Da mesma forma que a função `printf` é uma função que está em uma biblioteca padrão do C, `scanf` também está; a diferença é que `scanf` é uma função de entrada de dados.

Vamos usar um exemplo de uso, apresentado na Figura 5, a seguir:

```
1  #include <stdio.h>
2  int main(){
3      int idade;
4      int anoAtual;
5      int anoNascimento;
6      printf("Digite sua idade: ");
7      scanf("%d",&idade);
8      printf("Digite o ano atual: ");
9      scanf("%d",&anoAtual);
10     anoNascimento=anoAtual-idade;
11     printf("Oi,voce nasceu em %d\n",anoNascimento);
12     return 0;
13 }
```

Figura 5. Código-fonte com o uso de printf e scanf.

Esse programa pede ao usuário que ele digite sua idade e o ano atual e, com esses dados, ele calcula o ano do nascimento. Na linha 6, temos o uso da função `printf`, que mostra na tela o texto, informando ao usuário a solicitação do dado.

Em seguida, aparece o uso da função `scanf`. Quando o usuário digitar um valor e teclar enter, a linha 7 pode ser traduzida para: "armazene na variável idade o valor inteiro que foi digitado". A sintaxe básica da função `scanf` é a que está representada na Figura 6:

Figura 6. Sintaxe da função scanf.

Podemos, também, receber mais de um número no mesmo `scanf`, conforme pode ser visto no exemplo da Figura 7, a seguir:

```
1  #include <stdio.h>
2  int main()
3  {
4      int num1, num2;
5      printf("Insira dois numeros: ");
6      scanf("%d %d", &num1, &num2);
7      printf("Você digitou: '%d' e '%d'", num1, num2);
8  }
```

Figura 7. Código-fonte de um programa que soma dois valores, recebidos no mesmo scanf.

Códigos de formatação

Os códigos de formatação em C começam com % e permitem que as funções de entrada (`scanf`) e saída (`printf`) expressem os tipos de dados armazenados nas variáveis. Indicam o tipo do dado.

- %c: imprime o conteúdo da variável com representação ASCII.
- %d: imprime o conteúdo da variável com representação decimal com sinal.
- %u: imprime o conteúdo da variável com representação decimal sem sinal.
- %o: imprime o conteúdo da variável com representação octal sem sinal.
- %x: imprime o conteúdo da variável com representação hexadecimal sem sinal.
- %f: imprime o conteúdo da variável com representação com ponto decimal.
- %e: imprime o conteúdo da variável com representação em notação científica (exponencial).
- %g: formato geral, escolhe a representação mais curta entre %f e %e.

Além desses códigos de formatação em C, nós temos as sequências de escape que podem auxiliar para impressão de texto na tela de forma formatada. Dentre elas, os mais usados são.

- \n: quebra de linha
- \t: realiza uma tabulação horizontal
- \b: adiciona um "backspace" ao texto
- \r: volta o cursor para o começo da linha sem mudar de linha
- \a: emite um sinal sonoro
- \": quando você precisa, no seu texto, do uso de aspas duplas
- \': quando você , no seu texto, do uso de aspas simples
- \\: quando você precisa, no seu texto, do uso de barra invertida
- \0: caractere nulo

No código da Figura 8, a seguir, você pode ver o uso de códigos de formatação e sequências de escape.

```
1 #include <stdio.h>
2 int main ()
3 {
4     int a;
5     printf ("Digite um número: ");
6     scanf ("%d", &a);
7     printf ("\nO número digitado foi %d\n", a);
8     return (0);
9 }
```

Figura 8. Uso de código de formatação, sequência de escape e funções de entrada e saída.

Leituras recomendadas

PAES, R. B. *Introdução à Programação com a Linguagem C*. São Paulo: Novatec, 2016. 296p.

PINHEIRO, F. A. C. *Elementos de programação em C*. Porto Alegre: Bookman, 2012. 548p.

DICA DO PROFESSOR

As funções printf e scanf são funções existentes na biblioteca padrão stdio.h do C e são muito usadas em praticamente todos os programas que precisam de entrada e saída de dados.

Acompanhe o vídeo da Dica do professor, onde você terá orientações sobre esses conceitos e sobre como usar essas funções.

Conteúdo interativo disponível na plataforma de ensino!

EXERCÍCIOS

1) Considerando o seguinte código, qual o valor das variáveis x e y no final da execução?

```
1  #include <stdio.h>
2  int main(){
3      int x,y;
4      x = 17;
5      y = x;
6      x = 25;
7      printf("%d %d\n",x,y);
8      return 0;
9  }
```

A) X é 17 e y é 17.

B) X é 25 e y é 25.

C) X é 17 e y é 25.

D) X é 25 e y é 17.

E) Não é possível determinar os valores das variáveis.

2) Qual o valor da variável após a execução do seguinte código:

```
1  #include <stdio.h>
2  int main(){
3      int x = 18;
4   int resposta = x - 2;
5      printf("o valor da variavel resposta eh: %d\n", resposta);
6      return 0;
7  }
```

A) 18.

B) -2.

C) O valor da variável x é um endereço de memória.

D) Nada. Ocorre um erro, pois não é possível realizar uma operação do tipo x-2.

E) 16.

3) Qual a definição de uma função em linguagem de programação?

A) Uma função é um conjunto de instruções organizadas em forma de bloco e que têm um nome para que assim sejam utilizadas em um código-fonte.

B) Uma função é uma regra matemática utilizada em programação para realizar cálculos.

C) É uma regra para acesso de forma mais simples à memória do computador.

D) Uma função é a forma com que os usuários entram com os dados em um programa.

E) Uma função é uma entidade básica de um sistema computacional e existe somente em linguagens antigas.

4) Para que servem as funções printf e scanf?

A) São funções utilizadas para realizar operações de impressão de um arquivo em um impressora e acesso ao *scanner* conectado à máquina.

B) São funções para realização de operações matemáticas em um computador.

C) A função printf mostra um valor na tela do computador e a função scanf serve para armazenar valores dentro de uma variável.

D) A função printf armazena um valor em uma determinada variável, já a função scanf mostra um valor na tela.

E) São instruções de acesso à memória.

5) O que está faltando na linha 9 para que seja mostrado o seguinte texto: Soma:, seguido do valor da soma dos dois números digitados pelo usuário.

```
1  #include <stdio.h>
2  int main()
3  {
4      int num1, num2,soma;
5      printf("Insira dois numeros: ");
6      scanf("%d %d", &num1, &num2);
7      soma=num1+num2;
8      printf("Você digitou: '%d' e '%d'", num1, num2);
9      printf("Soma: %d",_);
10     return 0;
11 }
```

A) &soma.

B) soma.

C) "soma".

D) 'soma'.

E) +soma.

NA PRÁTICA

Ana está gerenciando o desenvolvimento de um sistema de pesquisa sobre o sistema bancário. O sistema é usado com interação direta com o usuário em terminais espalhados por um *shopping* para verificar o uso do aplicativo do banco e da confiabilidade dos clientes em cada banco disponível no terminal.

O sistema pede alguns dados para o usuário, como:

- Qual o banco utilizado?
- Você tem o aplicativo do banco instalado no celular?
- Qual o seu grau de escolaridade?
- Você confia no banco que utiliza?

As perguntas estão coladas ao lado do caixa eletrônico.

A ideia é avaliar as feições de comportamento, portanto, é avisado ao usuário que o seu rosto estará sendo filmado.

Analisando os dados e as feições do comportamento das pessoas, a equipe percebeu que os usuários tentam responder mas erram a ordem das respostas ou esquecem o que precisam responder, não finalizando o processo. E muitas vezes ficando chateados com o comportamento do sistema.

- Como Ana poderia melhorar a interação com o usuário desse sistema?

Como ela poderia fazer com que o usuário soubesse, dentro do próprio sistema, qual a pergunta que precisa responder e a ordem?

Depois de realizar algumas pesquisas, **Ana percebeu que a única coisa que precisava ser adicionada a cada pedido de dado que é feito usando a função scanf(), é uma linha de código mostrando a pergunta para o usuário, e para isso ela usou o printf().** Assim, quando for solicitado qual o banco que o usuário usa, por exemplo, será preciso um par de instruções como a seguinte:

```
printf("Qual o Banco que você utiliza: 1 - Banco X 2 - Banco Y ");
scanf("%d", &banco) ;
```

SAIBA MAIS

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

CS Principles: Defining and Calling Functions

Acompanhe no vídeo a seguir, como definir e usar funções em programação.

Conteúdo interativo disponível na plataforma de ensino!

Programação em C

Acesse o vídeo para compreender melhor o que são os subprogramas ou funções, pois o vídeo traz uma introdução sobre esse conceito.

Conteúdo interativo disponível na plataforma de ensino!

Programar em C - Formatação de Ouput -

Nesse vídeo você irá aprender formatação de output na Linguagem C.

Conteúdo interativo disponível na plataforma de ensino!